Amendments to the Drawings:

The attached sheet of drawings includes changes to Figure 1.

Attachment: Replacement Sheet

REMARKS

This Amendment is in response to the Office Action mailed October 17, 2005. In the Office Action, the disclosure and drawings were objected, claims 24-26 were rejected under 35 U.S.C. §112 and claims 1-26 were rejected under 35 U.S.C. §102(e). By way of the present response, Applicants have: 1) amended claims 1, 8, and 13; 2) canceled claims 15 and 24-26; and 3) added no new claims.

Drawing Objection

The drawings of the subject application were objected based on an incorrect reference label in FIG. 1. In response, Applicants have amended FIG. 1 to include the reference label "196" to be associated with thread 1. Withdrawal of the drawing objection is respectfully requested.

Claim Objection

Claim 18 was objected to as having an incorrect spelling for processor. Applicants have amended claim 18 in a non-narrowing manner to correct this typographical error. Withdrawal of the claim objection is respectfully requested.

Rejections Under 35 U.S.C. §112

Claims 24-26 were rejected under 35 U.S.C. §112 (first & second paragraphs). While Applicants disagree with these rejections, Applicants have canceled these claims to further prosecution.

Rejection Under 35 U.S.C. § 102(e)

Claims 1-26 were rejected under 35 U.S.C. §102(e) as being anticipated by Kalafatis (U.S. Patent No. 6,535,905).

Kalafatis describes "thread switching operation within a multithreaded processor" in certain situations. (See Kalafatis, Abstract.) For example, Kalafatis describes "that [when] the allocator [of a microprocessor] determines that insufficient resources ... are available for instructions (i.e., microinstructions) for a particular thread received from the queue ... the allocator 76 asserts a stall signal ...On the assertion of such a stall signal 82 for a particular thread, it may be desirable to perform a thread switching operation." (Kalafatis, Col. 13, lines 32-46.) Kalaftis is not describing relinquishing portions of thread partionable resources but merely that if there are insufficient resources available for a particular thread it may be desirable switch to a different thread.

Kalafatis further describes "[i]n a multithreaded processor, where a page miss occurs for an instruction stream of a current thread, it may be advantageous to perform a thread switching operation so as to allow an alternative thread to utilize the latency introduced by the page walk operation. ... [A] determination is made as to whether a predetermined minimum quantity of instruction information (e.g., a predetermined minimum number of chunks) for an alternative thread (e.g., thread 1) are pending and available for dispatch from the logical partition 124 of the instruction streaming buffer 106." (Kalafatis, Col. 18, line 62 through Col. 19, line 53.) Kalaftis is not describing relinquishing portions of thread partionable resources but determining if a minimum number of instruction information is available for an alternative thread to process.

With respect to claim 1, Kalafatis does not describe:

> a plurality of thread partitionable resources that are each partitionable between a plurality of threads including a first thread and at least one other thread;
>
> a plurality of shared resources shared by the plurality of threads including the first thread and at least one other thread; and
>
> logic to receive a program instruction from a first thread directing said processor to suspend execution of said first thread, and in response to said program instruction to cause the processor to suspend execution of the first thread and to relinquish portions of said plurality of thread partitionable resources associated with the first thread for use by other ones of said plurality of threads.

The Office Action equates the instruction queue 103, scheduler 72, and execution unit 70 as being "an overall partitionable resource" with, for example, the "execution portion of the resource [] relinquished so that it may be used by" the other thread. While the instruction queue 103 is described as being partitionable in Kalafatis, the scheduler 72 and execution unit 70 are not. The latter two components being shared resources. However, Kalafatis does not describe that thread switching causes the relinquishing portions of thread partitionable resources as required.

Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 1 requires. Claims 2-12 are dependent upon claim 1 and are allowable for at least the same reason.

With respect to claim 13, Kalafatis does not describe:

> receiving a first opcode in a first thread of execution;
> suspending said first thread for a selected amount of time in response to said first opcode;
> relinquishing a plurality of thread partitionable resources in response to said first opcode, wherein relinquishing said plurality of thread partitionable resources comprises:
> relinquishing a partition of an instruction queue; and
> relinquishing a plurality of registers from a register pool.

Kalafatis does not describe that thread switching causes the relinquishing a partition of an instruction queue.as required by the claim. Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 13 requires. Claims 14 and 16-17 are dependent upon claim 13 and are allowable for at least the same reason.

With respect to independent claim 18, Applicants respectfully submit that Kalafatis does not describe:

> a memory to store a plurality of program threads, including a first thread and a second thread, said first thread including a first instruction;

> a processor coupled to said memory being separate from said processor, said processor including a plurality of thread partitionable resources and a plurality of shared resources, said processor to execute instructions from said memory, said processor, in response to execution of said first instruction to suspend said first thread and to relinquish portions of said plurality of thread partitionable resources.

While the instruction queue 103 is described as being partitionable in Kalafatis, the scheduler 72 and execution unit 70 are not. The latter two components being shared resources. Kalafatis does not describe that thread switching causes the relinquishing portions of thread partitionable resources as required by the claim. Accordingly, Applicants respectfully submit Kalafatis does not describe what Applicants' claim 18 requires. Claims 19-23 are dependent upon claim 18 and are allowable for at least the same reason.

Conclusion

In view of the remarks made above, it is respectfully submitted that pending claims 1-26 define the subject invention over the prior art of record. Thus, Applicants respectfully submit that all the pending claims are in condition for allowance, and such action is earnestly solicited at the earliest possible date.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: August 7, 2006 By____________________________

Daniel M. DeVos
Reg. No. 37, 813

12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025